Санкт-Петербургский государственный университет
Кафедра моделирования социально-экономических систем

Крюков Константин Андреевич

Выпускная квалификационная работа бакалавра

Негладкая многошаговая задача оптимального распределения ресурсов

Направление 010400

Прикладная математика и информатика

Научный руководитель:
ассистент кафедры МСЭС
Парфенов А. П.

Санкт-Петербург

2017

Содержание

Введение

Успешное ведение хозяйства и бизнеса состоит из цепочки поэтапно принимаемых решений, основанных на различных законах рыночной экономики. От оптимально принятого единичного решения зачастую зависит весь конечный результат деятельности.

В конечном итоге, хозяйственная деятельность подчинена цели получения прибыли на основе рационального использования ресурсов. Необходимо постоянно соотносить зависимость выпуска определенного объема продукции и затрачиваемых производственных ресурсов.

Актуальным решением в данном случае становится выбор количества производственных ресурсов при заданных параметрах производственной функции, что и определяет эффективность использования ресурсов на результаты производственной деятельности.

На основе производственных функций ведется долгосрочное планирование на перспективу, с учетом различных факторов и вариантов развития. В этом смысле возможности аппарата производственных функций неисчерпаемы, поэтому изучение данной темы является актуальным.

В качестве производственных функций можно рассматривать CES—функции (это общий случай, который включает в себя функции Леонтьева, функции Кобба—Дугласа и линейные функции).

Чаще всего рассматриваются задачи распределения одной или нескольких продукции, а в качестве критериев оптимальности, которые следует максимизировать, берутся прибыль или обьём производства.

Подобные задачи решаются методами динамического программирования, для которых возможны различные алгоритмы точного и приближенного решения. Алгоритмы оптимизации дискретных функций и оптимизация непрерывных функций методом табуляции рассмотрены в работах Беллмана Р. [9].

В данной работе ставится многошаговая задача, которая усложняет задачу Беллмана: имеется несколько видов ресурсов, каждый из которых необходимо распределить на производство, которое определяется их производственными функциями, при этом остаток ресурсов(незадействованные ресурсы) нужно максимизировать.

Цель дипломной работы: Разработать негладкую многошаговую задачу оптимального распределения ресурсов.

Задачи дипломного исследования:

а) *Поставить задачу оптимального распределения ресурсов.*

б) *Рассмотреть необходимые и достаточные условия оптимальности для задачи распределения ресурсов.*

в) *Построение алгоритма оптимизации методом субградиента.*

г) *Привести примеры задач с использованием производственных CES—функций.*

Объект исследования: задача распределения ресурсов в экономических системах.

Предмет исследования: многошаговая выпуклая задача оптимизации с использованием производственных функций.

Методология и методы исследования. Используются методы решения задач гладкого нелинейного программирования, основанные на градиентах, и методы решения задач негладкого нелинейного программирования, основанные на субдифференциалах.

Работа состоит из введения, шести глав, примера, заключения и списка использованных источников.

1 Постановка задачи

Задано дискретное время $t_0 \dots, T$. Задано n различных ресурсов, пронумеруем их от 1 до n. Даны *технологические процессы* представляемые в виде производственных функций.

В данном исследовании использованы определение и три аксиомы из работы Симонова П.М. относительно производственных функций.

Определение. [3] Математическая модель в виде формулы зависимости выпуска продукции от вектора затрачиваемых или используемых в производстве ресурсов получила название *производственной функции* (ПФ).

Предприниматели заинтересованы в максимальном эффекте от использования ресурсов. Неэффективное использование ресурсов предприниматели и менеджеры предприятия, как правило, не допускают. Поэтому ПФ отражает зависимость максимально возможного выпуска продукции от ресурсов, введенных в производство. И естественно производственная функция не может быть отрицательной.

Аксиоматика моделирования производства в математической форме выражает общие свойства производства:

Аксиома 1.0.1. *[3] Каждый ресурс производится хотя бы одним технологическим процессом. Для этого на его ПФ подаются различные требуемые ресурсы(в нужно количестве).*

Аксиома 1.0.2. *[3] Если на вход технологического процесса подаются положительные количества ресурсов, то на выходе — положительное количество продукции.*

Аксиома 1.0.3. *[3] Про отсутствие рога изобилия: если на входе технологического процесса все количества нужных ресурсов равны 0, то производство отсутствует.*

Пусть имеется вектор количеств ресурсов $X(t)$ размерности n, $x_i \geqslant 0$, $x_i \in \mathbb{R}$ при $\forall i$. Каждому виду ресурсов ставится в соответствии вектор производственных функций:

$$f_i(Y(t)) = (f_{i1}(Y_{i11}(t), ...Y_{i1n}(t)), \ldots, f_{im_i}(Y_{im_i1}(t), ...Y_{im_in}(t)) \tag{1.1}$$

размерности m_i, где i—ресурс. $Y_{ij}(t)$ — вектор затрачиваемых или используемых в производстве ресурсов размерности n. Таким образом, каждая производственная функция зависит от вектора размерности n, и у каждой функции этот вектор свой. В каждый момент времени на вход каждой производственной функции подается свой вектор ресурсов. $Y_{ijk}(t)$ — количество ресурса k для j-го технологического процесса, производящего ресурс i. ПФ f_{ij} зависит от количеств ресурсов $y_{ij1}(t), ...y_{ijn}(t)$, причем на шаге t не возможно использовать ресурсов больше, чем имеется на данном этапе:

$$x_k(t) \geqslant \sum_{i=1}^{n} \sum_{j=1}^{m_i} y_{i,j,k}(t) \tag{1.2}$$

при $\forall k$.

В рассмотрение берутся производственные функции видов:

а) *Гладкие функции* для них написаны условия Куна—Таккера.

б) *Кусочно-гладкие функции* частным случаем которых являются *Кусочно-линейные функции* для этого случая построен алгоритм методом субградиентов.

В данной работе мы рассматриваем только выпуклые ПФ.

Мы предполагаем, что производство занимает 1 единицу времени, поэтому ресурсы, вложенные в производство в момент $t-1$, дают результат в момент t. Производственные функции возвращают количество i—ресурса

$$\sum_{j=1}^{m_i} f_{ij}(Y_{ij1}(t-1), ...Y_{ijn}(t-1)) = x_i(t). \tag{1.3}$$

Помимо этого существует потребление различных ресурсов,

$$\sum_{j=m_i} f_{ij}(Y_{ij1}(t-1), ...Y_{ijn}(t-1)) = x_i(t) \geqslant \sum_{l,j} y_{l,j,k=i}(t), \qquad (1.4)$$

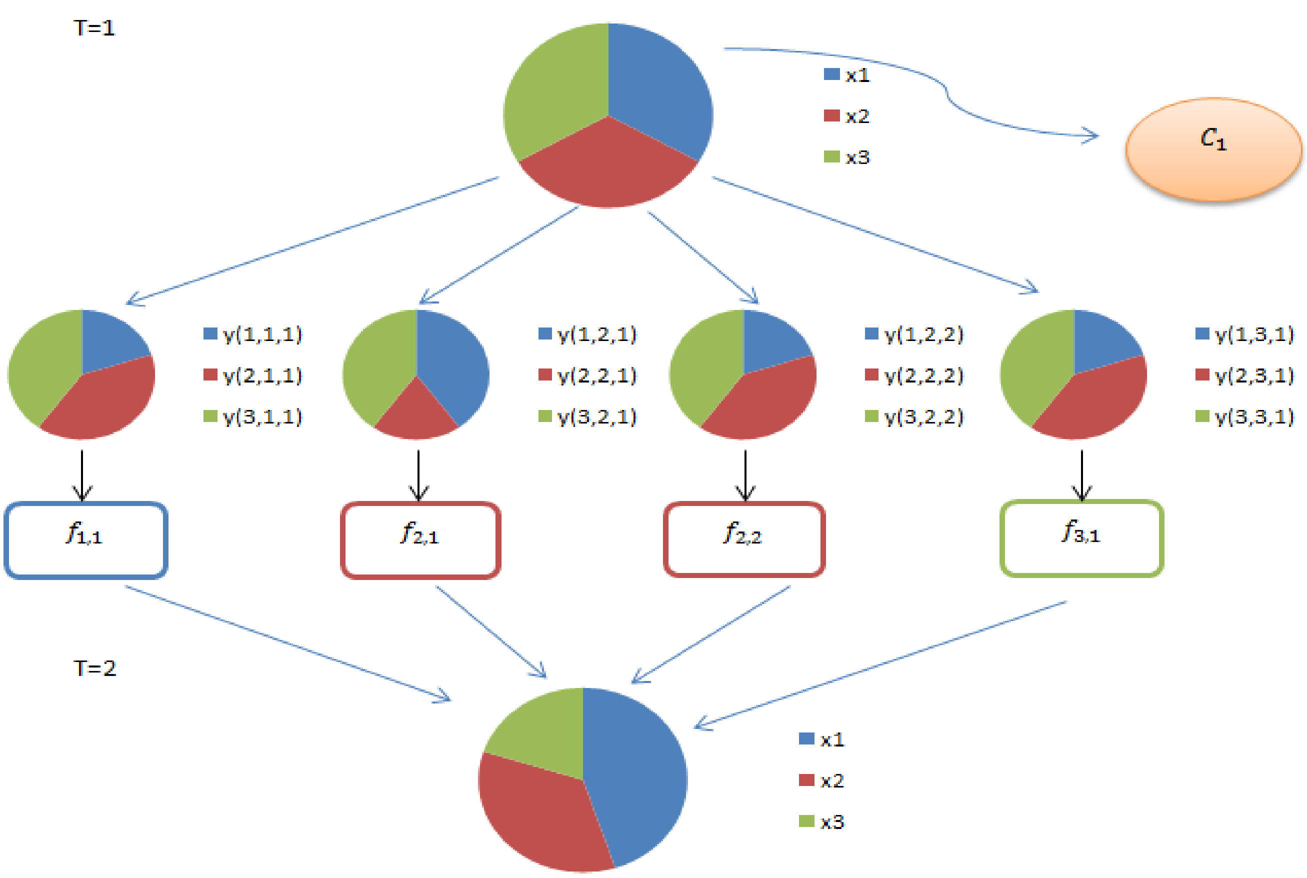
т.е. на каждом шаге ресурсов мы посылаем на технологический процесс не больше, чем у нас есть на данном шаге, из предыдущего неравенства следует, что

$$x_k(t) = c_k(t) + \sum_{i,j} y_{i,j,k}(t), x_k \geqslant 0, y_{i,j,k} \geqslant 0 \forall i,j,k \qquad (1.5)$$

,где i-получаемый ресурс, j-номер технологического процесса, k-ресурс, отправленный на производство.

$c_k(t)$ — потребление ресурсов в момент t, и отсюда мы его можем выразить как разность 1.5:

$$c_k(t) = x_k(t) - \sum_{i,j} y_{i,j,k}(t) \qquad (1.6)$$



На данном рисунке показана схема распределения ресурсов за 2 итерации, имеются три вида ресурсов, раскрашенные в три цвета. $f_{1,1}$ *и*

$f_{3,1}$ производят первый и третий ресурс соответственно, а $f_{2,1}$ и $f_{2,2}$ производят второй ресурс.

На каждую производственную функцию подается свой вектор ресурсов, оставшиеся ресурсы идут на потребление.

В результате итерации получаем вектор произведенных ресурсов всех производственных функций.

Необходимо максимизировать потребление ресурсов, а именно сумму $\sum_{t=1}^{T}\sum_k(c_k(t))$, путем оптимального распределения количества ресурсов на каждый технологический процесс за T итераций:

$$
\begin{aligned}
&max_Y\left(\sum_{t=1}^{T}(\sum_i(f_i(y_i(t-1))-\sum_{j,k}y_{ijk}(t)))\right),\\
&y_{ijk}(t)\geqslant 0,\\
&\sum_j(f_{i,j}(y_i(t-1)))-\sum_{k,j}y_{kji}(t)\geqslant 0\ ,\\
&\forall i,j,k;\ t=\overline{1,T}.
\end{aligned}
$$

Также нашу задачу можно переписать $\max(c(t))$ на $\min(-c(t))$, где $c(t)$-вогнутая целевая функция, а $-c(t)$- выпуклая.

2 Условия Куна—Таккера для задачи распределения ресурсов

Решаем задачу нелинейного программирования.

Данную задачу можно решить методом динамического программирования. Но, поскольку множество состояний системы непрерывно, функция Беллмана в каждый момент времени будет зависеть от непрерывного множества аргументов и иметь сложный кусочно заданный вид. Это сильно усложняет алгоритм.

Введем обозначения:

$$\phi(y) = \sum_{t=1}^{T}(\sum_i(f_i(y_i(t-1)) - \sum_{jk} y_{ijk}(t))$$
$$\psi_i(y) = \sum_j(f_{i,j}(y_i(t-1))) - \sum_{k,j} y_{kji}(t),$$
$$\forall i,j,k;\ t = \overline{1,T}.$$

Функцию $\phi(y)$ будем использовать как целевую функцию, а с помощью функций $\psi_i(y)$ будем задавать ограничения, где $i = \overline{1,n*T}$.

При рассмотрении выпуклой задачи максимизации $\phi(y)$ при условиях $\psi_i(y) \geqslant 0$ может случиться так, что не будет существовать таких $\lambda_j > 0$, $j = 1,...,n$, для которых без дополнительных предположений о природе функций были бы справедливы уравнения 2.2, 2.3, где $\bar{y}$ — оптимальное решение. Эти дополнительные предположения называют условиями регулярности ограничений.

Определение. *Условия регулярности Слейтера*

$$\exists \tilde{\bar{y}} \in Y | \forall i = \overline{1,n} : \psi_i(\tilde{\bar{y}}) > 0, \tag{2.1}$$

Утверждение 2.0.1. *Наша задача удовлетворяет условиям регулярности Слейтера*

Доказательство. Покажем, что существует хотя бы одна внутренняя точка допустимого множества решений.

Допустим, мы имеем положительный вектор начальных ресурсов $x_i(0) > 0$ для $\forall i$, и по Аксиоме 1.0.1 у каждого ресурса есть своя производственная функция. Разобьем каждый i-ресурс на равное количество частей $y_{i1} = ... = y_{in}$ для каждой производственной функции. Мы получим на следующий итерации по аксиоме 1.0.2 положительный вектор ресурсов $x_i(1) > 0$ для $\forall i$. Т.К. Задача непрерывна, мы можем бесконечно много продолжать данный алгоритм, следовательно, у нас будет хотя бы одна точка, лежащая внутри области наших ограничений.

Если мы введем потребление $c_i(t)$, которое строго меньше имеющихся на данный момент ресурсов, то для этого случая применяем уже разработанный алгоритм для ресурсов $x_i(t) - c_i(t)$.

$\square$

Функция Лагранжа (лагранжиан) этой задачи имеет следующий вид:

$$L(y,\lambda) = \lambda_0 \sum_{t=1}^{T} (\sum_{i} (f_i(y_i(t-1)) - \sum_{jk} y_{ijk}(t)) + \sum_{l=1}^{nT} (\lambda_l \psi_l(y)), \tag{2.2}$$

где коэффициенты λ_l называют *множителями Лагранжа.*

Теорема 2.0.2. *Обратная теорема Куна—Таккера (достаточное условие оптимальности):*

Пусть Функции $\phi(y)$, $\psi_1(y),...,\psi_n(y)$ дифференцируемы и вогнуты, множество Y выпукло и точка $\bar{y}$ допустима в задаче , причем $\bar{y} \in (Y)$.

Пусть, кроме того, существуют множители Лагранжа $\lambda_j > 0$, $j = 1,...,n$, такие что при $\lambda_0 = 1$ выполнены следующие соотношения:

$$\frac{\delta L(\bar{y},\lambda)}{\delta y_i} = 0;\ \ i = \overline{1,n} \tag{2.3}$$

и

$$\sum_{j=1}^{n} \frac{\delta L(\bar{y},\lambda)}{\delta \lambda_j} \lambda_j = 0; \tag{2.4}$$

Тогда $\bar{y}$ - решение задачи.

Условия Куна-Таккера применимы к нашей задаче, если производственные функции - гладкие.

3 Субградиенты

Рассмотрим еще один метод выпуклой минимизации.

Определение ([4], с. 160)**.** Пусть f — выпуклая функция. Вектор g — называется *субградиентом* функции f в точке x_0, если для любого x выполняется неравенство:

$$f(x) \geqslant f(x_0) + (g, x - x_0); \tag{3.1}$$

Множество $\partial f(x_0)$ всех субградиентов для f в x_0 называется *субдифференциалом функции* f в точке x_0 [4].

Теорема 3.0.1 ([4], с. 163)**.** *Равенство* $f(x^*) = min_x f(x)$ *имеет место тогда и только тогда, когда*

$$0 \in \partial f(x^*) \tag{3.2}$$

Выпуклая функция имеет локальный минимум в x^* тогда и только тогда, когда 0 принадлежит субдифференциалу функции в этой точке.

Введенные субградиенты будут в дальнейшем применяться при построении алгоритмов максимизации. Тем не менее для применения этих схем при реализации необходимо быть уверенным, что субградиенты могут быть правильно посчитаны. Рассмотрим несколько правил таких вычислений:

Лемма 3.0.1.1 ([4], с. 163)**.** *Пусть функция* f *выпукла и замкнута. Предположим, что она дифференцируема на всей своей области определения. Тогда* $\partial f(x) = \{f'(x)\}$ *для всех* x *.*

Свойства:

а) $\partial\left(\lambda f_1(x)\right) = \lambda \partial f_1(x)$

б) $\partial\left(f_1(x) + f_2(x)\right) = \partial\left(f_1(x)\right) + \partial\left(f_2(x)\right)$

Теорема 3.0.2 ([6], с. 73). *Пусть* $f_1, ..., f_m$ *— выпуклые функции на линейном векторном пространстве* X*, непрерывные в точке* x_0*, а*

$$f(x) = max_{1 \leq i \leq m} f_i(x). \tag{3.3}$$

Введем набор активных в точке x_0 *индексов:*

$$J = \{i \in \{1, ..., m\} \mid f_i(x_0) = f(x)\}. \tag{3.4}$$

Тогда

$$\partial f(x_0) = co\{\cup_{i \in J} \partial f_i(x_0)\}, \tag{3.5}$$

4 Производственная CES-функция

Функция CES (CES — англ. Constant Elasticity of Substitution) — функция, обладающая свойством постоянной эластичности замещения, применяемая в экономической теории.CES — функция применяется в первую очередь для моделирования производственной функции. Иногда она используется также и для моделирования функции полезности. Некоторые другие популярные производственные функции представляют собой частные или предельные случаи данной функции.

$$f(x_1,\ldots,x_n) = A\left(\sum_{i=1}^{n}\lambda_i x_i^{\rho}\right)^{k/\rho} \tag{4.1}$$

где x_i - количество i-го ресурса, k - показатель однородности, ρ - эластичность замещения.

Функция Кобба-Дугласа - частный случай CES-функции при $\rho \to 0$

$$A\prod_{i=1}^{n} x_i^{\lambda_i} \tag{4.2}$$

Если сделать предельный переход $\rho \to -\infty$ функции:

$$f(x_1,\ldots,x_n) = (\sum_{i=1}^{n}\lambda_i x_i^{\rho})^{1/\rho}, \tag{4.3}$$

То получим:

$$f(x_1,\ldots,x_n) = \min_i(\lambda_i x_i), \tag{4.4}$$

Для определения интервалов выпуклости(вогнутости)CES-функции, рассмотрим *неравенства Минковского*:

$$M_p(x) = \left(\frac{1}{n}\sum_i x_i^p\right)^{1/p} \tag{4.5}$$

Утверждение 4.0.1. *Функция $M_p(x)$ является вогнутой функцией, если $p \leqslant 1$ и выпуклая функция, если $p \geqslant 1$.*

Этот результат следует из *неравенства Минковского.*

Результат работ над неравенством Минковского изложены([8], стр. 19-20, [10], стр. 31):

Теорема 4.0.2. *Пусть* $f(x)$,$g(x) \geqslant 0$ *и* $p < 1 (p \neq 0)$, *тогда для* неравенства Минковского *следует:*

$$\left(\int (f(x) + g(x))^p dx \right)^{1/p} \geqslant \left(\int f(x)^p dx \right)^{1/p} + \left(\int g(x)^p dx \right)^{1/p}$$

Из Теоремы 4.0.2 следует, что CES-функция 4.1 вогнута при $p < 1$.

Аналогично доказывается [[7], с. 368], что CES-функция 4.1 выпукла $p > 1$

Нашу задачу можно переписать $\max(c(t))$ на $\min(-c(t))$, где $c(t)$-вогнутая целевая функция, а $-c(t)$- выпуклая.

Теорема 4.0.3. *([4] стр. 161.) Пусть функция* $f(x)$ *выпукла и замкнута. Тогда* $\partial f(x_0)$ *— непустое ограниченное множество.*

Утверждение 4.0.4. *Сумма субдифференцируемых функций субдифференцируема.*

Теорема 4.0.3 и утверждение4.0.4, говорят нам, что у целевой функции нашей задачи, которая использует CES-функция при $p \leqslant 1$ существует субдифференциал.

5 Субградиентный метод

Определение. [4] Субградиентным методом называется процесс составления последовательности $\{x_k\}_{k=0}^{\infty}$ по правилу:

$$x_{k+1} = x_k - h_k \frac{g_f(x_k)}{\parallel g_f(x_k) \parallel} \tag{5.1}$$

где h_k—шаг, $g_f(x_k)$— субградиент $f(x)$ в x_k, если $g_f(x_k) = 0$, то x_k точка минимума и построение останавливается

Регулировка шага $h_{k+1} = h_k * q_k$, где $0 < q_k < 1$, $k = 0{,}1{,}2{,}...$

Метод включает в себя применения процедуры наискорейшего спуска в направлении антисубградиента выпуклой функции.

Остановка цикла происходит по условию $\parallel g_f(\bar{x}_k) \parallel \leqslant \varepsilon_g$,где $\bar{x}_k \in [x_k{,}x_{k+1}]$

В приложение А реализован субградиентный метод минимизации выпуклых функций.

Данный алгоритм является безусловной оптимизацией, для анализа множества ограничений используется метод штрафов.

В данной примере используется метод внутренней точки.

6 Примеры задач для случая с использованием производственных CES-функций

Рассмотрим пример задачи, где имеется два вида ресурсов x_1,x_2 и даны две производственные функции:

$$f_1(y_{1,1,1},y_{1,1,2}) = \left(\lambda_1 y_{1,1,1}^{\rho} + \lambda_2 y_{1,1,2}^{\rho}\right)^{1/\rho}$$

$$f_2(y_{2,1,1},y_{2,1,2}) = \left(\lambda_1 y_{2,1,1}^{\rho} + \lambda_2 y_{2,1,2}^{\rho}\right)^{1/\rho}$$

Напишем задачу максимизации для двух функций:

$$max_Y \Bigg(\sum_{t=1}^{T} (f_1[y_{1,1,1}(t-1),y_{1,1,2}(t-1)] + f_2[y_{2,1,1}(t-1),y_{2,1,2}(t-1)] -$$
$$-(y_{1,1,1}(t) + y_{1,1,2}(t) + y_{2,1,1}(t) + y_{2,1,2}(t))) \Bigg) \ ,$$
$$y_{1,1,1}(t) \geqslant 0 \ ,$$
$$y_{1,1,2}(t) \geqslant 0 \ ,$$
$$y_{2,1,1}(t) \geqslant 0 \ ,$$
$$y_{2,1,2}(t) \geqslant 0 \ ,$$
$$f_1[y_{1,1,1}(t-1),y_{1,1,2}(t-1)] - (y_{1,1,1}(t) + y_{2,1,1}(t)) \geqslant 0 \ ,$$
$$f_2[y_{2,1,1}(t-1),y_{2,1,2}(t-1)] - (y_{1,1,2}(t) + y_{2,1,2}(t)) \geqslant 0 \ ,$$
$$t = \overline{1,T}.$$

Введем обозначения:

$$\phi(y) = \sum_{t=1}^{T} (f_1[y_{1,1,1}(t-1),y_{1,1,2}(t-1)] + f_2[y_{2,1,1}(t-1),y_{2,1,2}(t-1)] -$$
$$-(y_{1,1,1}(t) + y_{1,1,2}(t) + y_{2,1,1}(t) + y_{2,1,2}(t))) \ ,$$
$$\psi_i(y) = f_1[y_{1,1,1}(t-1),y_{1,1,2}(t-1)] - (y_{1,1,1}(t) + y_{2,1,1}(t)) \ ,$$
$$\psi_j(y) = f_2[y_{2,1,1}(t-1),y_{2,1,2}(t-1)] - (y_{1,1,2}(t) + y_{2,1,2}(t)) \ ,$$
$$t = \overline{1,T}.$$

Пусть нам нужно максимизировать потребление за $T = 3$—итерации. Не уменьшая общности примим: $y_1 = y_{1,1,1}$ $y_2 = y_{1,1,2}$ $y_3 = y_{2,1,1}$ $y_4 = y_{2,1,2}$ Перепишем задачу:

$$max_Y\left(\sum_{t=1}^{3}(f_1[y_1(t-1),y_2(t-1)] + f_2[y_3(t-1),y_4(t-1)] - \right.$$

$$\left. -(y_1(t) + y_2(t) + y_3(t) + y_4(t)))\right) \ ,$$

$$y_1(t) \geqslant 0 \ ,$$

$$y_2(t) \geqslant 0 \ ,$$

$$y_3(t) \geqslant 0 \ ,$$

$$y_4(t) \geqslant 0 \ ,$$

$$f_1[y_1(0),y_2(0)] - (y_1(1) + y_3(1)) \geqslant 0 \ ,$$

$$f_2[y_3(0),y_4(0)] - (y_2(1) + y_4(1)) \geqslant 0 \ ,$$

$$f_1[y_1(1),y_2(1)] - (y_1(2) + y_3(2)) \geqslant 0 \ ,$$

$$f_2[y_3(1),y_4(1)] - (y_2(2) + y_4(2)) \geqslant 0 \ ,$$

$$f_1[y_1(2),y_2(2)] - (y_1(3) + y_3(3)) \geqslant 0 \ ,$$

$$f_2[y_3(2),y_4(2)] - (y_2(3) + y_4(3)) \geqslant 0 \ ,$$

Перепишем обозначения:

$$\phi(y) = \sum_{t=1}^{3}(f_1[y_1(t-1),y_2(t-1)] + f_2[y_3(t-1),y_4(t-1)] -$$

$$-(y_1(t) + y_2(t) + y_3(t) + y_4(t))) \ ,$$

$$\psi_1(y) = f_1[y_1(0),y_2(0)] - (y_1(1) + y_3(1)) \ ,$$

$$\psi_2(y) = f_2[y_3(0),y_4(0)] - (y_2(1) + y_4(1)) \ ,$$

$$\psi_3(y) = f_1[y_1(1),y_2(1)] - (y_1(2) + y_3(2)) \ ,$$

$$\psi_4(y) = f_2[y_3(1),y_4(1)] - (y_2(2) + y_4(2)) \ ,$$

$$\psi_5(y) = f_1[y_1(2),y_2(2)] - (y_1(3) + y_3(3)) \ ,$$

$$\psi_6(y) = f_2[y_3(2),y_4(2)] - (y_2(3) + y_4(3)) \ ,$$

$$L(y,\lambda) = \lambda_0 \sum_{t=1}^{3}(f_1[y_1(t-1),y_2(t-1)] + f_2[y_3(t-1),y_4(t-1)] -$$

$$-(y_1(t) + y_2(t) + y_3(t) + y_4(t))) + \sum_{l=1}^{6}(\lambda_l \psi_l(y)),$$

Распишем Лагранжиан:

$$L(y,\lambda) = \lambda_0(f_1[y_1(0),y_2(0)] + f_2[y_3(0),y_4(0)] + f_1[y_1(1),y_2(1)]+$$
$$+f_2[y_3(1),y_4(1)] + f_1[y_1(2),y_2(2)] + f_2[y_3(2),y_4(2)] - (y_1(1) + y_2(1) + y_3(1)+$$
$$+y_4(1) + y_1(2) + y_2(2) + y_3(2) + y_4(2) + y_1(3) + y_2(3) + y_3(3) + y_4(3)))+$$
$$+\lambda_1(f_1[y_1(0),y_2(0)] - (y_1(1) + y_3(1))) + \lambda_2(f_2[y_3(0),y_4(0)] - (y_2(1) + y_4(1)))+$$
$$+\lambda_3(f_1[y_1(1),y_2(1)] - (y_1(2) + y_3(2))) + \lambda_4(f_2[y_3(1),y_4(1)] - (y_2(2) + y_4(2)))+$$
$$+\lambda_5(f_1[y_1(2),y_2(2)] - (y_1(3) + y_3(3))) + \lambda_6(f_2[y_3(2),y_4(2)] - (y_2(3) + y_4(3))),$$

Применим теорему 2.0.2:

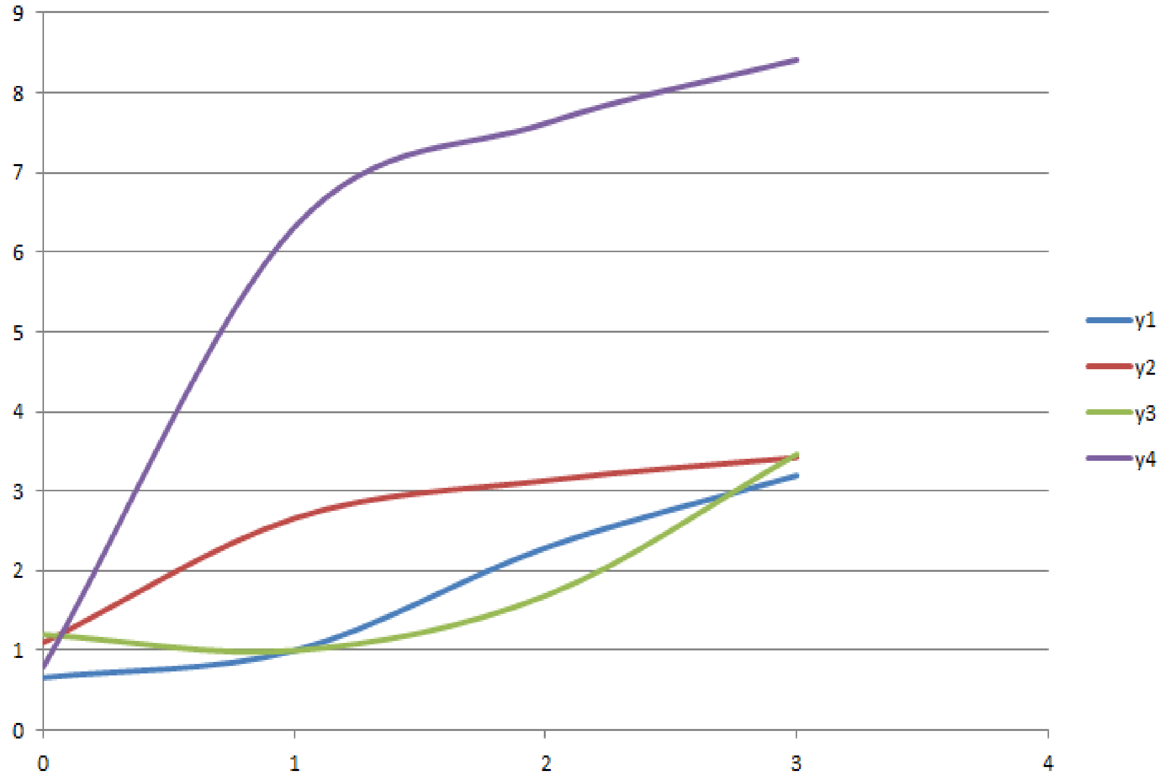
$$\frac{\delta L(y,\lambda)}{\delta y_1} = \lambda_0\left(\frac{\delta f_1[y_1(0),y_2(0)]}{\delta y_1} + \frac{\delta f_1[y_1(1),y_2(1)]}{\delta y_1} + \frac{\delta f_1[y_1(2),y_2(2)]}{\delta y_1}\right)-$$
$$-(1+1+1) + \lambda_1\left(\frac{\delta f_1[y_1(0),y_2(0)]}{\delta y_1} - 1\right) + \lambda_3\left(\frac{\delta f_1[y_1(1),y_2(1)]}{\delta y_1} - 1\right)+$$
$$+\lambda_5\left(\frac{\delta f_1[y_1(2),y_2(2)]}{\delta y_1} - 1\right) = 0;$$
$$\frac{\delta L(y,\lambda)}{\delta y_2} = \lambda_0\left(\frac{\delta f_1[y_1(0),y_2(0)]}{\delta y_2} + \frac{\delta f_1[y_1(1),y_2(1)]}{\delta y_2} + \frac{\delta f_1[y_1(2),y_2(2)]}{\delta y_2}\right)-$$
$$-(1+1+1) + \lambda_1\left(\frac{\delta f_1[y_1(0),y_2(0)]}{\delta y_2}\right) + \lambda_3\left(\frac{\delta f_1[y_1(1),y_2(1)]}{\delta y_2}\right)+$$
$$+\lambda_5\left(\frac{\delta f_1[y_1(2),y_2(2)]}{\delta y_2}\right) - (\lambda_2 + \lambda_4 + \lambda_6) = 0;$$
$$\frac{\delta L(y,\lambda)}{\delta y_3} = \lambda_0\left(\frac{\delta f_2[y_3(0),y_4(0)]}{\delta y_3} + \frac{\delta f_2[y_3(1),y_4(1)]}{\delta y_3} + \frac{\delta f_2[y_3(2),y_4(2)]}{\delta y_3}\right)-$$
$$-(1+1+1) + \lambda_2\left(\frac{\delta f_2[y_3(0),y_4(0)]}{\delta y_3}\right) + \lambda_4\left(\frac{\delta f_2[y_3(1),y_4(1)]}{\delta y_3}\right)+$$
$$+\lambda_6\left(\frac{\delta f_2[y_3(2),y_4(2)]}{\delta y_3}\right) - (\lambda_1 + \lambda_3 + \lambda_5) = 0;$$
$$\frac{\delta L(y,\lambda)}{\delta y_4} = \lambda_0\left(\frac{\delta f_2[y_3(0),y_4(0)]}{\delta y_4} + \frac{\delta f_2[y_3(1),y_4(1)]}{\delta y_4} + \frac{\delta f_2[y_3(2),y_4(2)]}{\delta y_4}\right)-$$
$$-(1+1+1) + \lambda_2\left(\frac{\delta f_2[y_3(0),y_4(0)]}{\delta y_4} - 1\right) + \lambda_4\left(\frac{\delta f_2[y_3(1),y_4(1)]}{\delta y_4} - 1\right)+$$
$$+\lambda_6\left(\frac{\delta f_2[y_3(2),y_4(2)]}{\delta y_4} - 1\right) = 0;$$

Аналогично можно расписать второе соотношение Теоремы Куна—Таккера, только при этом все выражения нужно просуммировать и изменить δy_j на $\delta\lambda_j$ и домножить на λ_j.

$$\sum_{j=1}^{n} \frac{\delta L(\bar{y},\lambda)}{\delta\lambda_j}\lambda_j = 0;$$

Решая полученную систему, получаем $\bar{y}$ —вектор максимума.

Покажем решение этой задачи вторым методом. Пусть в данном примере $p = 2$. Для данной задачи написана программная реализация на языке MATLAB, использующая субградиентный метод. При написании программы использовалась данная литература [5]



Ось абсцисс показывает количество итераций, а ось ординат количество ресурсов y_i.

Рассмотрим второй пример задачи, где имеется также два вида ресурсов x_1,x_2 и даны две производственные функции:

$$f_1(y_{1,1,1}, y_{1,1,2}) = \min(\lambda_1 y_{1,1,1}, \lambda_2 y_{1,1,2})$$

$$f_2(y_{2,1,1}, y_{2,1,2}) = \min(\lambda_1 y_{2,1,1}, \lambda_2 y_{2,1,2})$$

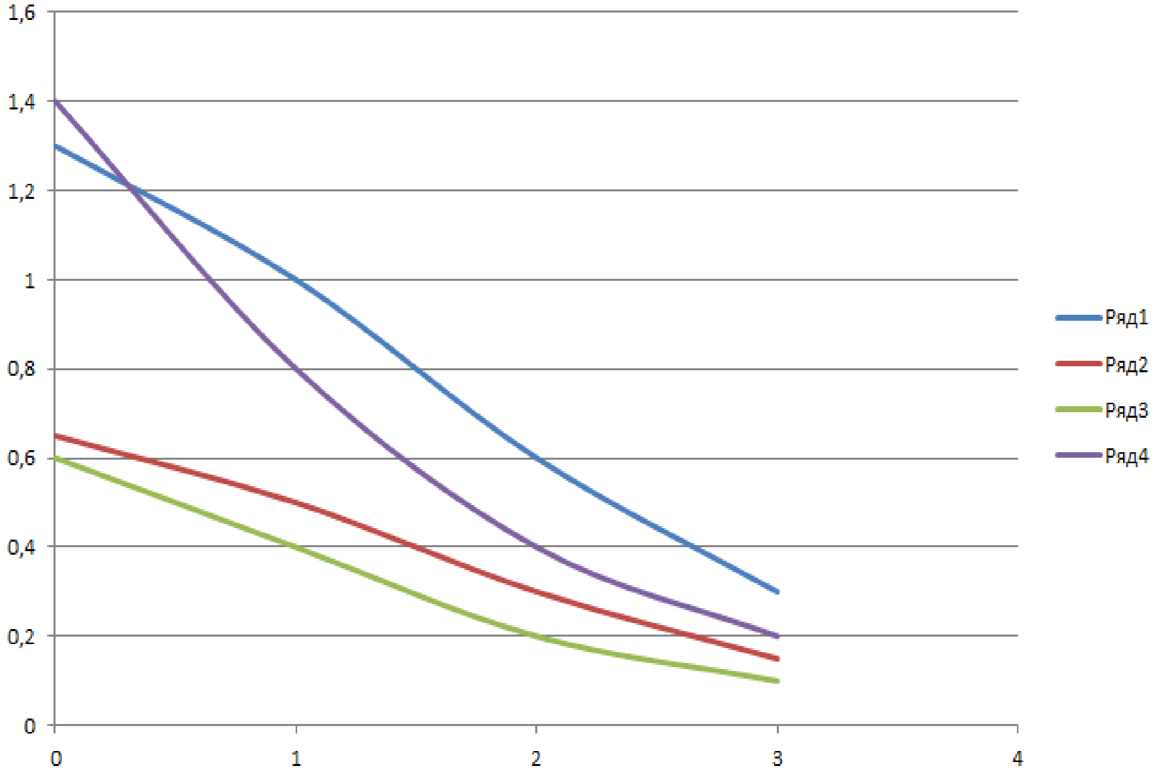
Для $T = 3$—итерации задача максимизация будет выглядеть также, как и в прошлом примере. В программу будем подавать функцию:

$$min_Y - \Bigg(\sum_{t=1}^{3} (f_1[y_1(t-1), y_2(t-1)] + f_2[y_3(t-1), y_4(t-1)] -$$
$$-(y_1(t) + y_2(t) + y_3(t) + y_4(t))) \Bigg)$$

В данном случае представленные функции имеют кусочно-гладкий вид.

В главе посвященной субградиентам, показывалось как выглядит субдифференциал таких функций.

Результаты программы субградиетного метода:



Ось абсцисс показывает количество итераций, а ось ординат количество ресурсов y_i.

На графике видно, что ресурсы подаются равномерно в зависимости от своих коэффициентов, а один из ресурсов пропадает из-за специфики функции, поэтому график убывает.

Заключение

В работе была рассмотрена задача оптимального распределения ресурсов в системах производства.

Проанализирована литература по данной теме, на ее основе выбраны методы для решения, а также разработана программная реализация алгоритмов, решающих задачу.

В процессе исследования, была расширенна задача Беллмана Р, добавлением негладких функций, а также изменена целевая функция, которая доставляет максимум потребления, заданному на множестве ресурсов данной системы.

В первой части были описаны построения данной задачи максимизации и основные свойства производственных функций для использования в выпуклой задачи оптимизации.

Во второй части были предложены вывод условий Куна—Таккера для вогнутых функций и сформулированы необходимые и достаточные условия существования максимума для этой системы, а также рассмотрен субградиентный метод.

В заключительной части для данных методов написана программная реализация на языке Matlab и проведено тестирование на примерах с использованием CES-функции.

Список использованных источников

1. *Зангвилл У.* Нелинейное программирование. Единый подход. М., “Советское радио”, 1973. 312 с.

2. *Карманов В.Г.* Математическое программирование: учеб. пособие. - М.: ФИЗМАТЛИТ, 2000.

3. *Симонов П.М.* Экономико-математическое моделирование.

4. *Нестеров Ю. Е.* Методы выпуклой оптимизации. Издательство МЦНМО, Москва;2010.

5. *Шор Н.З* Методы минимизации недифференцируемых функций и их приложения:Киев, 1970. 44 с.

6. *Стрекаловский А. С.* Введение в выпуклый анализ: учеб. пособие. – Иркутск: Иркут. ун-т, 2009. – 81 с.

7. *Ильин В. А. Садовничий В. А. Сендов Бл. Х.* Математический анализ. Начальный курс,— 2-е изд., перераб., — М.: Изд-во МГУ, 1985. — 662 с.

8. *Beckenbach EF, Bellman R* Inequalities. Springer, Berlin-Gottingen; 1961.

9. *Bellman R* Динамическое программирование,Издательство иностранной литературы,1960.

10. *Hardy GH, Littlewood JE, Polya G* Inequalities. Cambridge University Press, Cambridge; 1934.

Приложение А Программный код реализации алгоритмов

```
%алгоритм реализован на языке программирования MATLAB
%maxitn, epsx , epsg — Критерии остановки
%x(n) — начальная точка
%calcfg (n, f ,g,x) — программа для вычисления f и субградиетна g
%itn — число затраченных итераций
%xr(n) — точка максимума
%fr — значения функции


function [xr,fr,itn,nc,stop] = Proga(x,alpha,h0,q1,q2,nh,epsg,
epsx,maxitn)
itn = 0; hs = h0; B = eye(length(x));
xr=x;
nc = 1; [fr,g0] = calcfg(xr);

size(g0);
g0
norm(g0)

if(norm(g0) < epsg) stop = 2;
    return;
end

for (itn = 1:maxitn)
  g1 = B' * g0;
  g1 = g1 / norm(g1);
```

```
dx = B * g1;
d = 1; ls = 0; ddx = 0;
while (d > 0)
   x =x- hs * dx;
   ddx =ddx+ hs * norm(dx);
   nc=nc+1 ;
   [f, g1] = calcfg(x);
   if (f < fr)
       fr = f;
       xr = x;
   end
   norm(g1)
       if(norm(g1) < epsg)
           stop = 2;
           return;
       end
       ls=ls+1;
           if(ls==nh)
               hs=hs* q2;
               ls=0;
           end


           if(ls > 500)
               istop = 5;
               return;
           end
               d = dx' * g1;
end
if(ls == 1)
```

```
      hs =hs* q1;
  end



     return;
 end
     dg = B’ * (g1 - g0);
     xi = dg / norm(dg);
     B = B+(1 / alpha - 1) * B * xi * xi’ ;
     g0 = g1;
 end
     stop = 4;

    fr
end




    \newpage
    %алгоритм реализован на языке программирования MATLAB
    % для простого случая линейной функции
function [v,s] = prost(A,x,y,z)
n=length(x);
for l = 0.1:0.1:0.9
    l
    d(1)=sqrt(l);
    for i=2:1:n
        d(i)=sqrt(1-l)/(n-1);
    end
```

```
for i=1:1:n
    c(i)=d(i)/(z(i)*x(i));
    a(i)=d(i)*(x(i)-y(i))/x(i);
    q(i)=y(i);
end
for k=1:1:4
    S=0;
    e=0;
    for i=1:1:n
        b(i)=a(i)*c(i);
    end
        j=1;
        while j<n+1
            S=1;
            e=b(1);
            for i=2:1:n
               if b(i)>e
                 e=b(i);
                 S=i;
               end
            end
             b(S)=0;
             g(j)=e;
             t(j)=S;
             j=j+1;
            end
             for j=1:1:n
                 s(j)=0;
                 b(j)=0;
```

```
     for i=1:1:j
          s(j)=s(j)+1/(c(t(i))*(10^3))^2;
          b(j)=b(j)+a(t(i))/c(t(i));
     end
     b(j)=(b(j)-A(k))/s(j)/(10)^6;
end
j=n;
p=0;

while g(j)<b(j)
     if j-1==0
          p=1;
          j
          break;
     else
             j=j-1;
             S=0;
     end

end
if p==0
if g(j)>=b(j)
     e=b(j);
     r=j;
     S=0;
end

for i=1:1:r
     v(t(i))=a(t(i))/c(t(i))-(e/c(t(i)))/c(t(i));
```

```
                S=S+v(t(i));
            end
            for i=r+1:1:n
                v(t(i))=0
            end
                v
                S
            for i=1:1:n
                q(i)=q(i)+v(i)/z(i);
                a(i)=d(i)*(x(i)-q(i))/x(i);
            end
            q
             end

      end
  end
```